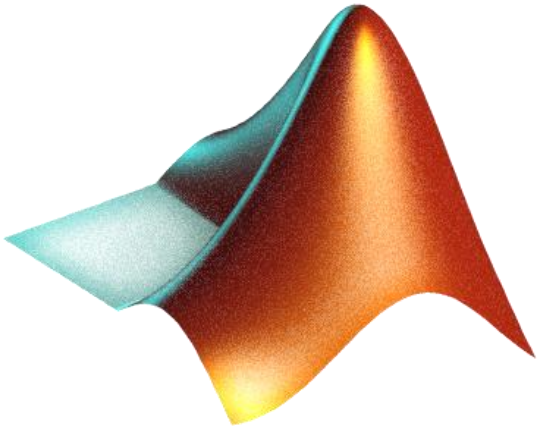
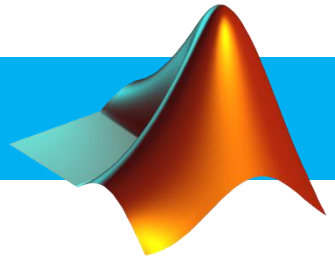
Curso Básico de MATLAB

Jorge De Los Santos

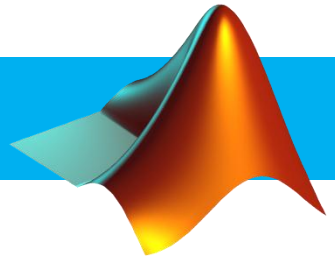


LAB DLS



Contenido del curso

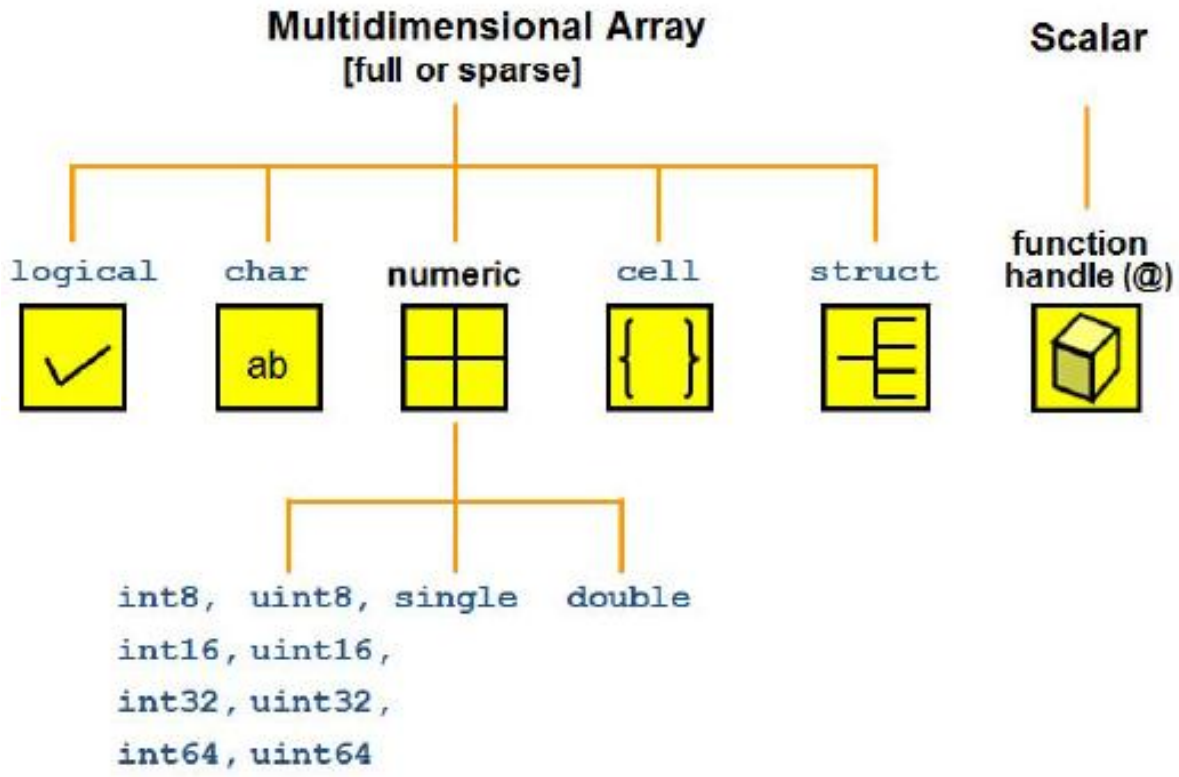
1. Introducción al entorno de desarrollo
2. Tipos de datos y operadores
3. Ficheros de comandos / scripts
4. Matrices y vectores
5. Variables simbólicas, manipulación algebraica y cálculo.
6. Gráficas en 2D
7. Gráficas en 3D
8. Entradas y salidas en el Command Window
9. Sentencias de control (if, elseif, else, swicth-case)
10. Bucles / Ciclos (for, while)
11. Sentencias try-catch, break, pause, continue y return
12. Funciones y sub-funciones
13. Cell arrays



Contenido del curso

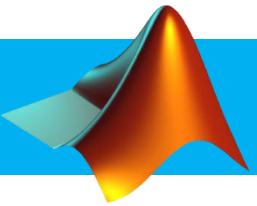
14. Estructuras
15. Exportar e importar variables del workspace
16. Exportar e importar datos de un fichero
17. Exportar e importar datos de Microsoft Excel
18. Análisis básico de datos
19. Manipulación de archivos y directorios
20. Solución de ecuaciones diferenciales
21. Introducción al procesamiento de imágenes
22. Introducción a las animaciones
23. Introducción al desarrollo de GUIs
24. Introducción a la POO
25. Depurar y exportar código
26. Compilar aplicaciones (crear ejecutables)

Tipos de datos



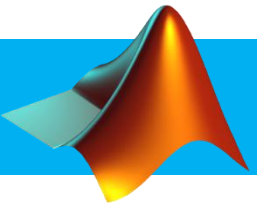
Operadores aritméticos

Operator	Description
+	Addition
-	Subtraction
.*	Multiplication
./	Right division
.\	Left division
+	Unary plus
-	Unary minus
:	Colon operator
.^	Power
.'	Transpose
'	Complex conjugate transpose
*	Matrix multiplication
/	Matrix right division
\	Matrix left division
^	Matrix power



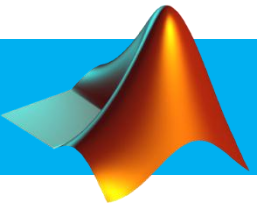
Operadores relacionales

Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to



Operadores lógicos

Operator	Description	Example
&	Returns 1 for every element location that is true (nonzero) in both arrays, and 0 for all other elements.	A & B = 01001
|	Returns 1 for every element location that is true (nonzero) in either one or the other, or both arrays, and 0 for all other elements.	A | B = 11101
~	Complements each element of the input array, A.	~A = 10010
xor	Returns 1 for every element location that is true (nonzero) in only one array, and 0 for all other elements.	xor(A,B) = 10100



Ficheros de comandos

Crear un nuevo script:

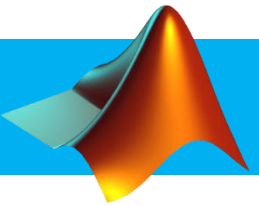
New>Script

O bien teclear en el Command Window:

```
>> edit
```

Editar un script existente:

```
>> edit nombre_script
```



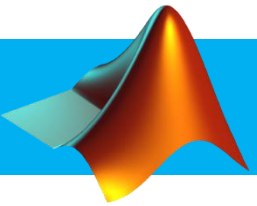
Matrices y vectores

Crear una matriz de 2x2:

```
>> A=[1 2;3 4]

A =

     1     2
     3     4
```

Crear un vector fila:

```
>> A=[3 -1 0]

A =

     3    -1     0
```



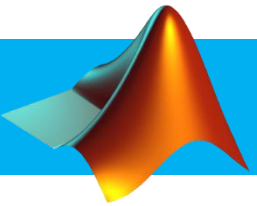
Variables simbólicas

Declarar una variable simbólica:

```
>> x=sym('x');
```

Declarar múltiples variables simbólicas:

```
>> syms x y z
```



Manipulación algebraica

Insertar expresiones algebraicas:

```
>>  x=sym('x');
>> (x+2)^2;
>> sin(x)+cos(x);
```

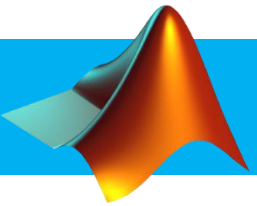
Factorizar y expandir expresiones algebraicas:

```
>> factor(x^2+2*x+1)

ans =

(x + 1)^2

>> expand((x+1)^2)

ans =

x^2 + 2*x + 1
```

Manipulación algebraica

Resolver ecuaciones:

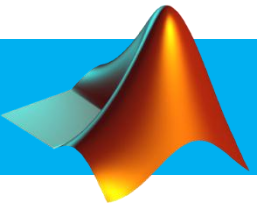
```
>> solve(x+2==3,x)

ans =

1

>> solve(x^2+2*x-1==0,x)

ans =

   2^(1/2) - 1
 - 2^(1/2) - 1
```



Manipulación algebraica

Resolver un sistema de ecuaciones:

```
>> sol=solve(x+y==1,x-y==0,x,y)

sol =

    x: [1x1 sym]
    y: [1x1 sym]

>> sol.x

ans =

1/2

>> sol.y

ans =

1/2
```



Cálculo diferencial

Límite de una función:

```
>> limit(sin(x)/x,x,0)
>> limit(1/(x-3),x,3)
```

Límites laterales:

```
>> limit(1/(x-3),'left')
>> limit(1/(x-3),'right')
```

Manipulación algebraica

Resolver ecuaciones:

```
>> solve(x+2==3,x)
>> solve(x^2+2*x+1==0,x)
```

Resolver un sistema de ecuaciones:

```
>>solve()
```

Manipulación algebraica

Resolver ecuaciones:

```
>> solve(x+2==3,x)
>> solve(x^2+2*x+1==0,x)
```

Resolver un sistema de ecuaciones:

```
>>solve()
```